Pattern Recognition Letters 30 (2009) 1059–1066

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Human action recognition by feature-reduced Gaussian process classification

Hang Zhou [a,*], Liang Wang [b], David Suter [a]

[a] Department of Electrical and Computer Systems Engineering, Monash University, Vic. 3800, Australia
[b] Department of Computer Science and Software Engineering, The University of Melbourne, Vic. 3010, Australia

ARTICLE INFO

Article history:
Available online 1 April 2009

Keywords:
Human action recognition
Characteristic-based descriptor
Gaussian process classification
Spectral feature reduction

ABSTRACT

This paper presents a spectral analysis-based feature-reduced Gaussian Processes (GP) classification approach to recognition of articulated and deformable human actions from image sequences. Using Tensor Subspace Analysis (TSA), space–time human silhouettes extracted from action sequences are transformed to a low dimensional multivariate time series, from which structure-based statistical features are extracted to summarize the action properties. GP classification, based on spectrally reduced features, is then applied to learn and predict action categories. Experimental results on two real-world state-of-the-art datasets show that the GP classification outperforms a Support Vector Machine (SVM). In particular, spectral feature reduction can effectively eliminate the inconsistent features, while leaving performance undiminished. Moreover, compared with Automatic Relevance Determination (ARD), the spectral way for feature reduction is more efficient.

1. Introduction

Human action recognition aims to understand patterns of human movements from video sequences, by classifying actions into known categories such as walking or jumping. This growing interest is strongly driven by a wide range of promising applications such as visual surveillance and human–machine interfaces. However, due to the great variation in the captured human action sequences (with respect to different instances or different persons: with various body types, action styles and speeds), how to extract distinguishable features to describe actions and to accurately model and recognize the actions, still remains a challenge.

'State-space' approaches using probabilistic models such as Hidden Markov Models (HMMs) (Brand et al., 1996; Nguyen et al., 2005) and Conditional Random Fields (CRFs) (Sminchisescu et al., 2005) have been widely used to model human action patterns. However, such probabilistic models are generally of high computational complexity, since detailed statistical modeling is usually required. Moreover, these involve assumptions about the probability distributions of variables of the dynamical models and the consequent development of inference methods as well as model parameter learning algorithms.

In contrast, being a kernel-based non-parametric model, the Gaussian process (GP) (Rasmussen and Williams, 2006) is more computationally tractable. General properties of the GP kernel are controlled by a few hyperparameters which can be estimated under the Bayesian framework. Furthermore, GP is used as a Bayesian prior to express beliefs about the underlying functions being modeled, which is linked to data via the likelihood. The posterior distribution can be directly calculated given the training data.

So far, the solutions applying GP to human action analysis include Wang et al. (2008), Raskin et al. (2007) and Zhou et al. (2008). Gaussian Process Dynamical Models (GPDM) are proposed in (Wang et al., 2008) for non-linear time series analysis with application to human action capture data. It is essentially a 'state-space' solution which models both the distribution of the observed data and the dynamics in the latent space. Raskin et al. (2007) proposed to combine GPDM and annealed particle filtering for tracking and classifying human actions. GP classification for human action recognition was first investigated in (Zhou et al., 2008) with better results than using a Support Vector Machine (SVM).

However, an issue for GP classification (as well as other classifications) is that each of the extracted input features is usually not guaranteed to have positive effect on the classification result. Including more input features could ultimately lead to poor classification outcomes. Only those features that are most relevant or beneficial to the prediction outputs should be selected. Neal (1996) use Automatic Relevance Determination (ARD) to assign each input a weight value (ranging from 0 to 1). The weight values have independent Gaussian prior distributions with standard deviation given by the corresponding hyperparameter with some prior. Then, the posterior distribution of the hyperparameters is calculated given the training data. The values of the hyperparameters are proportional to the corresponding input weight values. An issue for ARD is: what prior should be used for the hyperparameters? (as it will have a significant impact on the accuracy of

* Corresponding author. Tel.: +61 3 99053454.
E-mail addresses: hang.zhou@eng.monash.edu.au (H. Zhou), lwwang@csse.unimelb.edu.au (L. Wang), d.suter@eng.monash.edu.au (D. Suter).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.03.013

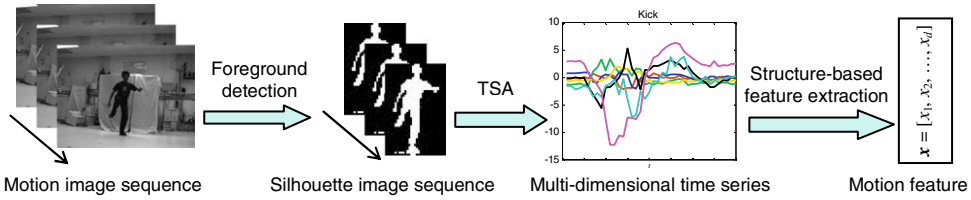

Fig. 1. From action image sequence to characteristic-based descriptor.

estimation results of the hyperparameters). Alternatively, Zhou and Suter apply spectral feature pruning in (Zhou and Suter, 2008). In that approach, the most homogeneous features (with similar 'effective dwell time') are retained and the rest discarded. This was the first time feature selection using spectral techniques has been applied to GPs. In essence, the (actually anisotropic) data have been modified to better match the (isotropic) kernel. However, that still leaves the problem of how to fit better to GP kernel for those data whose features have the same 'effective dwell time'.

To address this problem, this paper uses an additional measure called "major bandwidth".[1] In our approach, characteristic-based feature descriptors (Wang et al., 2008; Wang et al., 2006) are first used to transform time-varying dynamic features of varied-length action sequences to fixed-length feature vectors (and thus the temporal classification problem is converted to a static classification one), which enables the use of GP. Spectral analysis is then applied to the extracted features where the most consistent features with similar "major bandwidth" are chosen as the inputs for GP classification. The "major bandwidth" measure makes it possible to choose the most homogeneous features among those with similar "effective dwell time" as described in (Zhou and Suter, 2008). Compared with ARD, the spectral way for feature selection avoids the need to choose priors for the hyperparameters that exists in ARD (Neal, 1996). Extensive experimental and comparative results on two recent action datasets demonstrate the effectiveness of our approach.

In addition, we have also chosen Nearest-Neighbor (NN) and Support Vector Machine (SVM) to implement baseline experiments. Since NN is one of the simplest and most common machine learning algorithms and SVM is the state-of-the-art kernel-based learning method, they are the appropriate methods to be compared with GP.

The remainder of this paper is structured as follows. Action feature extraction is introduced in Section 2 and a brief review of GP classification is given in Section 3. Section 4 describes the feature reduction algorithm. In Section 5, experimental results are presented and discussed, prior to a summary of main conclusions of the work in Section 6.

2. Feature extraction of actions

Given a database consisting of v action sequences $M = \{M_1, M_2, \ldots, M_n\}$, we extract informative space–time silhouettes to represent the actions performed. The process of feature extraction is illustrated in Fig. 1. For each action sequence M_i including T_i image frames, i.e., $M_i = \{I_1^i, I_2^i, \ldots, I_{T_i}^i\}$, $i = 1, 2, \ldots, n$, the associated sequence of human silhouettes can be obtained by foreground detection techniques. Since the size and position of the foreground region vary among different frames, we construct normalized silhouettes $S_i = \{S_1^i, S_2^i, \ldots, S_{T_i}^i\}$ by centering the silhouette images and normalizing them to the same dimension of $n_1 \times n_2$.

To represent human actions in a more compact subspace rather than in the high dimensional image space, we adopt a structured dimensionality reduction method, i.e., Tensor Subspace Analysis (TSA) (He et al., 2005), to perform subspace learning of the articulated action space. TSA preserves the spatial information of silhouette images by representing an image as a second-order tensor (or a matrix). Given a set of m normalized silhouette images from the training data $\{S_1, S_2, \ldots, S_m\}$ in $R^{n_1} \otimes R^{n_2}$, TSA aims to find two transformation matrices U of size $n_1 \times l_1$ and V of size $n_2 \times l_2$ that map these silhouette images to another set $\{Y_1, Y_2, \ldots, Y_m\}$ in $R^{l_1} \otimes R^{l_2}$ $(l_1 < n_1, l_2 < n_2)$, such that $Y_i = U^T S_i V$. For more details about TSA, the reader may refer to He et al. (2005). In the learned tensor subspace, any silhouette sequence S_i can be accordingly projected into a trajectory $P_i = \{P_1, P_2, \ldots, P_{T_i}\}, P_i \in R^{l_1} \otimes R^{l_2}$. We can regard P_i as a form of multivariate time series with the number of dimensions $l = l_1 \times l_2$.

Structure-based statistical features are then extracted to summarize the multivariate time series: which turns action time series of different lengths into a feature vector of the same length. The nine most informative, representative and easily measurable characteristics are chosen to summarize the time series structure (Wang et al., 2006): *trend*, *seasonality*, *serial correlation*, *non-linearity*, *skewness*, *kurtosis*, *self-similarity*, *chaotic*, and *periodicity*. Based on these characteristics, corresponding metrics are calculated to form the structure-based feature vectors (Wang et al., 2008), called characteristic-based descriptors. For each dimension of P_i, we obtain 13 statistical features. Thus, a multi-dimensional time series P_i is summarized by a δ-dimensional ($d = 13 \times l$) feature vector $\boldsymbol{x}$.

3. GP classification

A Gaussian processes (GP) is a collection of random variables, any finite number of which has a joint Gaussian distribution (Rasmussen and Williams, 2006). A GP is fully specified by its mean function $m(\boldsymbol{x})$ and kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$, expressed as:

$$f \sim GP(m, k). \tag{3.1}$$

The GP classification process models the posterior directly. The GP prior is represented by the kernel function which characterizes correlations between points in the training data (which is a sample process). The kernel function's hyperparameters can be learned from the training data.

The kernel function used in this paper is the Radial Basis Function (RBF), also called the Squared Exponential (SE) function or Gaussian function (Snelson, 2006), i.e.,

$$k_{\mathrm{RBF}}(x - x') = \sigma_0^2 \exp\left[-\frac{1}{2}\left(\frac{x - x'}{l}\right)^2\right], \tag{3.2}$$

where x and x' are input pairs, l is the characteristic length scale and σ_0 is the signal variance. RBF is an isotropic kernel whose main advantage over non-isotropic versions is the simplicity. RBF is shift and rotation invariant on both signal and frequency domains as shown in Fig. 2. The isotropy of the RBF kernel is the fundamental property for feature reduction (see Section 4).

[1] This paper is actually an extended version of our previous work described in (Zhou et al., 2008).

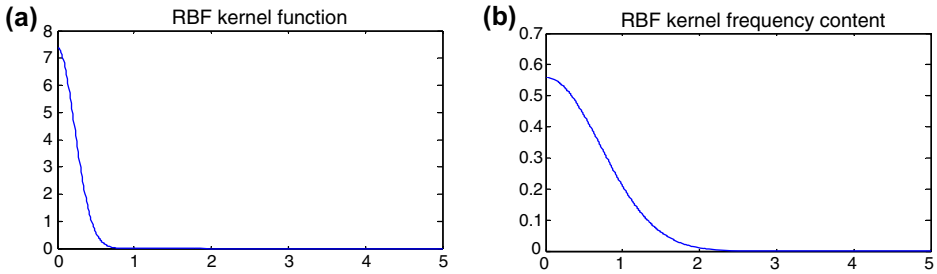

Fig. 2. (a) RBF kernel function (one feature/dimension). (b) Corresponding RBF kernel frequency content.

Assume that we have a dataset D with N observations $D = \{(x_i, y_i)|i = 1, \ldots, N\}$, where x is the input vector of dimension d (e.g., action feature vector here) and y is the class labels $+1/-1$. The input $N \times d$ matrix is denoted as X. Predictions for new inputs x^* are made out of this given training data using the GP model. As described in (Rasmussen and Williams, 2006), GP binary classification is done by first calculating the distribution over the latent function f corresponding to the test case,

$$p(f_*|X, y, x_*) = \int p(f_*|X, x_*, f)p(f|X, y)df, \tag{3.3}$$

where

$$p(f|X, y) = p(y|f)p(f|X)/p(y|X) \tag{3.4}$$

is the latent variable posterior. $p(f^*|X, x^*, f)$ is the predictive posterior w.r.t. possible latent functions, and the values of this could lie anywhere within the range of $(-\infty, +\infty)$. So the probabilistic prediction is made by

$$\bar{\pi}_* = p(y_* = +1|X, y, x_*) = \int \sigma(f_*)p(f_*|X, y, x_*)df_* \tag{3.5}$$

where the sigmoid function σ 'squashes' the prediction output to guarantee the valid probabilistic value within the range of $[0, 1]$.

For multi-class classification problem, we can treat each one class as being independent from the others, and apply binary classification individually to each (one) class versus the rest classes (although this may not be an optimal way to generalize to many classes).

4. Feature reduction

The input features (i.e., data on different dimensions of feature space) could vary significantly from each other while the GP RBF kernel is an isotropic kernel as shown in Eq. (3.2). Therefore, it appears to be not suitable to apply the RBF kernel blindly to the data. In this paper, the issue of how to better fit the anisotropic data to the isotropic kernel is analyzed from a spectral point of view, by using the "major bandwidth" measure defined in Section 4.2. Only the spectrally consistent features are chosen to match with the isotropic GP kernel's frequency content as illustrated in Fig. 2 and the rest discarded.

4.1. Non-uniform data spectral analysis

In order to analyze the high dimensional and unevenly spaced data on each dimension individually and efficiently, the high dimensionality of the unevenly spaced training input is reduced by applying the non-uniform discrete Fourier transform (NDFT) (Bagchi and Mitra, 1999) to the data on each dimension (feature) separately. For a N samples training dataset $D = \{(x_i, y_i)|i = 0, \ldots, N-1\}$, where x is the input vector of d-dimensions and y is the class labels $+1/-1$, the frequency content on each dimension is calculated as

$$F_{mf} = \sum_{n=0}^{N-1} y_n e^{-j2\pi f x_n}, \quad m = 0, \ldots, d-1, \tag{4.1}$$

where f is the frequency value on the frequency domain. The calculated NDFT magnitude $|F_m|$ is called "frequency content" hereafter.

Then we employ the notion of "effective dwell time" from Bretthorst's work (Bretthorst, 2000). It is one of the characteristic values of non-uniform data. (In our case, the signals are defined over feature space dimensions which are not normally time related. However, we use here, for the purposes of exposition, the term "time" to generally describe the input variable. See below for further details.)

As stated in (Bretthorst, 2000), for uniformly sampled data in time domain, let N be the total number of samples, $freq_k$ being the frequency. Uniform samples occur at

$$t_j = j\Delta T, \quad j \in \{0, 1 \ldots N-1\}, \tag{4.2}$$

where ΔT is the time interval between data samples, called 'dwell time'. Frequencies are then given by

$$freq_k = \frac{k}{N\Delta T}, \quad k \in \left\{-\frac{N}{2}, -\frac{N}{2}+1, \ldots, \frac{N}{2}\right\}. \tag{4.3}$$

The bandwidth is given as

$$-freq_{N_c} \leqslant freq \leqslant freq_{N_c}. \tag{4.4}$$

where the frequency $freq_{N_c}$ is called Nyquist critical frequency and given by

$$freq_{N_c} = \frac{1}{2\Delta T}. \tag{4.5}$$

In non-uniform data, there is no ΔT such that all the acquisition times are integer multiples of this value. We assume that (approximately), there exists a largest "effective dwell time" $\Delta T'$ where all of the times satisfy

$$t_l = k_l \Delta T', \tag{4.6}$$

where k_l is an integer. $\Delta T'$ is always less than or equal to the smallest time interval between data items. Eq. (4.6) is also true for uniformly sampled data where $k_l = 0, 1, \ldots, N-1$.

The "effective dwell time" $\Delta T'$ is used to define the Nyquist critical frequency

$$freq_{N_c} = \frac{1}{2\Delta T'}. \tag{4.7}$$

In our application, data is collected in space domain. Therefore, simply substitute time with feature data location values. Eq. (4.7) implies that it is the data location intervals that fully determine the frequency range of NDFT.

"Effective dwell time" $\Delta T'$ can be estimated using Eq. (4.6) where $\Delta T'$ can be regarded as the maximum common divisor of all the times t_l. In our application, t_l is equivalent to the feature data

locations (i.e., the high dimensional input data on each dimension) and $\Delta T'$ is related to the location difference. t_l is typically not an integer and being normalized to [0 1] in our application. Therefore, a practical and approximate way to implement Eq. (4.6), i.e. finding maximum common divisor for non-integer data locations is described in Algorithm 1.

The "effective dwell time" on each dimension $\Delta T'_i,\ i = 1, \ldots, d$ is thus obtained, where d is the number of features. The bandwidth can be calculated from the "effective dwell time" using Eqs. (4.4) and (4.7).

Algorithm 1. "Effective dwell time" (bandwidth) estimation for non-integer data

(1) Calculate R which represents how well all t_l can be divided by $l_{\min}$, i.e., the average modulus ratio of all t_l over $l_{\min}$

$$R = \frac{1}{N}\sum_{l=1}^{N}\frac{\mathrm{mod}(t_l, l_{\min})}{t_l}, \tag{4.8}$$

where N is the total number of data points.

(2) When R is smaller than a preset threshold which is set as $R_{thres} = 0.01$ here, $l_{\min}$ is taken as the common divisor of all t_l, i.e.
If

$$R < R_{thres}. \tag{4.9}$$

Then

$$\Delta T' = l_{\min}. \tag{4.10}$$

Else

$$l_{\min} = l_{\min} - 1. \tag{4.11}$$

(3) Repeat steps (1) and (2) until Eq. (4.9) is satisfied.
(4) "Effective dwell time" $\Delta T'$ is estimated following Eq. (4.10).

4.2. Feature reduction

In (Zhou and Suter, 2008), the consistency among features is defined as having the similar "effective dwell time". However, "effective dwell time" only defines the frequency range of the data frequency content. For features with the same "effective dwell time", the relevant data frequency content wave could still differ significantly from each other which give rise to the necessity of further refined measures on the frequency content characteristics. "Major bandwidth (*mbw*)" is one of such measures.

"Major bandwidth" is the bandwidth where the major energy of the data frequency content is concentrated and class label y is a determining factor on it. "Major bandwidth" is defined as: In a low pass data frequency content, the range of frequency from zero to the smallest frequency whose data frequency content value is below the preset threshold. Since GP kernels all have low pass frequency content, choosing the feature space based on "major bandwidth" is an effective way to pick out the most consistent features and feed to the isotropic GP kernel so as to optimize the GP classification.

An example of "major bandwidth" is shown in Fig. 3a where F is the data frequency content (as defined in Section 4.1). *mbw* should be the smallest frequency value satisfying

$$F(f) > \alpha * \max(F), \quad f < mbw, \tag{4.12}$$

$$F(f) \leqslant \alpha * \max(F), \quad f = mbw, \tag{4.13}$$

where α is the threshold for extracting the central lobe and filtering the noisy side lobes. It is empirically set to be 0.25 here. For a Gaussian distribution, this covers around 90% of the values or drawn within 1.6 standard deviation from the mean.

Fig. 4 is a further illustration of "major bandwidth" using synthetic data in Fig. 4a. The sampled signal and the corresponding data frequency content are shown in Fig. 4c and b respectively. Samples in Fig. 4e are obtained by swapping the class labels of the two samples in Fig. 4c without changing the sample intervals (which means the "effective dwell time" keeps unchanged). It can be seen in Fig. 4d that the shape of the data frequency content (which could be measured by the "major bandwidth") obviously changed although the "effective dwell time" remains the same.

In order to estimate *mbw*, the data bandwidth, i.e., the "effective dwell time" should be calculated first as described in Section 4.1.

mbw is then estimated for each feature in the following way,

(1) Find out the frequency content peak value max(F) within the bandwidth.
(2) Locate *mbw* value following Eqs. (4.12) and (4.13).

In this way, the major bandwidth values mbw_i, $i = 1, \ldots, d$ are estimated. To extract features with similar *mbw* which can be better modeled by the isotropic GP kernels, the following algorithm is implemented:

(1) Cluster mbw_i values into two parts using fuzzy c-means (i.e. degree of membership are given).
(2) In the larger cluster of the two, pick out the most correlated features which are those with the uppermost fuzzy membership function values (within the range of [0,1]), so that the picked out features account for a preset percentage of the total feature number. We set this percentage to be 50%, i.e. we pick out 50% of the most correlated features for our test in this paper. This is a reasonable preset threshold, as for most of the real-world data which is sub-Gaussian, a large part of the corresponding features are isotropic.
(3) Retain the picked out features and discard the rest.

5. Experiments and results

5.1. Human action data processing and classification

We use two state-of-the-art datasets, i.e., Dataset I (Veeraraghavan et al., 2006) and Dataset II (Blank et al., 2005), to evaluate our approach. Dataset I consists of 100 video sequences from 10 different actions performed by one subject (i.e., pick up object, jog in place, push, squash, wave, kick, bend to the side, throw, turn around, and talk on cell phone), and 10 different instances for each action. Different instances of the same action may consist of varying relative speeds, so this dataset is used to examine the effect of the temporal rate of execution on action recognition. Some sample images are shown in Fig. 5a.

Dataset II includes 90 low-resolution videos from 9 different people, each performing 10 actions in an *repetitive*manner (i.e., bend, jump jack, jump forward-on-two-legs, jump-in-place-on-two-legs, run, skip, gallop-sideways, walk, wave-one-hand, and wave two-hands)[2]. From these 90 videos where each video is a mixture of several repetitive same actions, we extract 198 sequences so

[2] http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html.

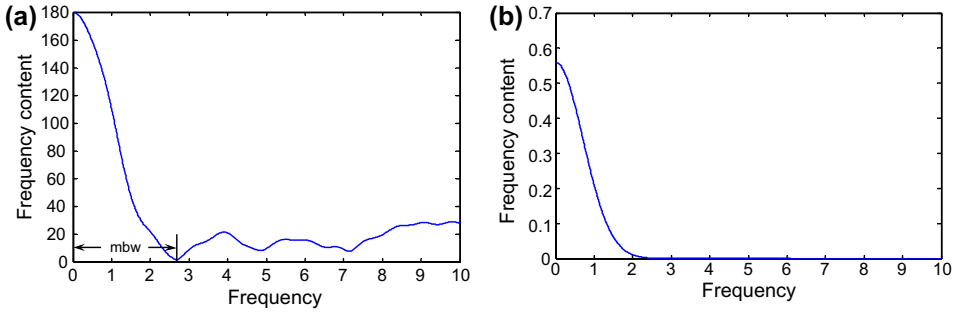

Fig. 3. (a) Data frequency content (one feature/dimension). (b) Corresponding RBF GP kernel frequency content.

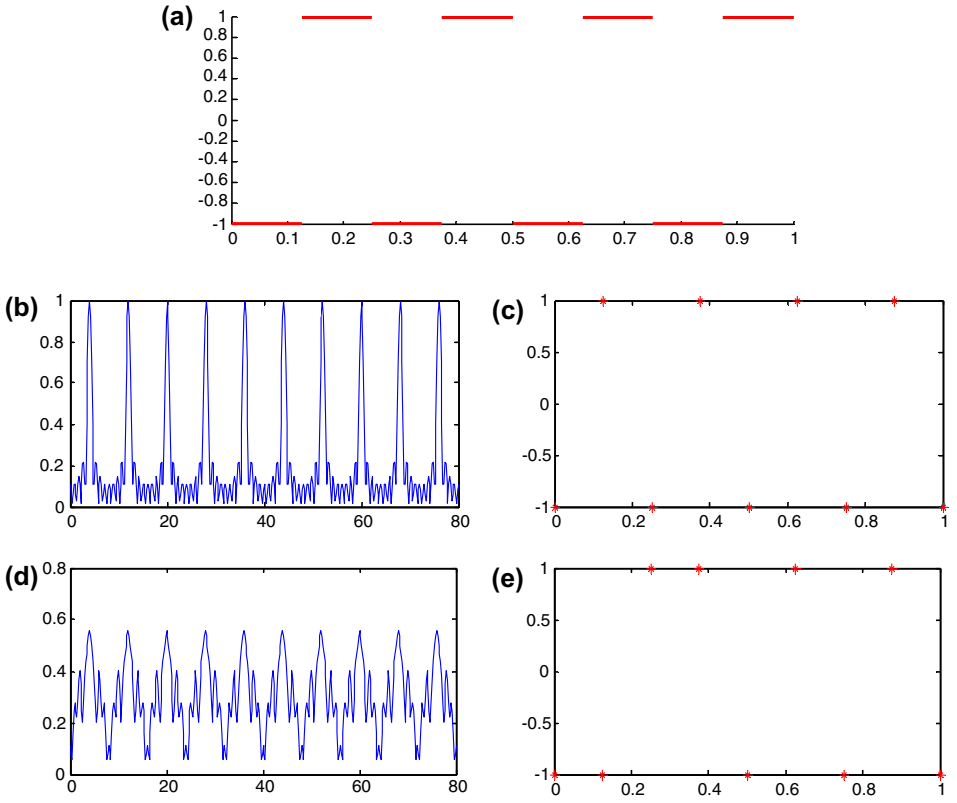

Fig. 4. (a) Original data signal. (b) Sampled data frequency content. (c) Sampled data. (d) Data frequency content after class label changed. (e) Data samples after class label changed.

that each of which includes a complete cycle of action. There exist inter-person differences between the same actions due to different physical sizes and action styles (and speeds), so this dataset is more realistic for the test of the method's versatility in terms of action variations at both temporal and spatial scales. Some sample images are shown in Fig. 5b.

Datasets I and II are provided with silhouette masks. We directly center and normalize these silhouette images into 48×32 resolution (i.e., $n_1 = 48$ and $n_2 = 32$). We then perform TSA (He et al., 2005) for dimensionality reduction. The reduced dimension of the input features is set to 4×4 (i.e., $l_1 = l_2 = 4$). That is, each action sequence is projected into a 16-dimensional time series in the embedding space, from which we extract 13 statistical features for each univariate time series (Wang et al., 2008). Then these features from each univariate time series are joined as one 208-dimensional (i.e., 16×13) vector.

To make the results comparable with that in (Wang et al., 2008), the data is processed in a similar way. For Dataset I, we divide the 100 video sequences into 10 disjoint sets with each of the set including one sample sequence of all the 10 distinct actions. For Dataset II, the number of sample sequences for the 10 actions are 9; 23; 24; 27; 14; 22; 25; 16; 19 and 19 respectively. This dataset is divided into 9 non-overlapping set in a way to make sequences of each action distribute evenly among the partitioned sets and each of them includes sequences of all the 10 actions. This is a more random and easy to choose division compared with the way in (Wang et al., 2008) where each of the nine divided set includes all actions from one person.

For either Dataset I or Dataset II, classification is done in a leave-one-out (LOO) way among the divided subsets, *i.e.*, each time leave one subset out (denote as *the test set*) for testing and all the remaining subsets (denote as *the training set*) for training. Three kinds of GP classification which we call GP, GP_ARD and GP_FO are implemented. GP applies the standard GP classification to each of the LOO training and testing set pair individually and takes the mean recognition rate as the final result. GP_ARD uses ARD (Neal,

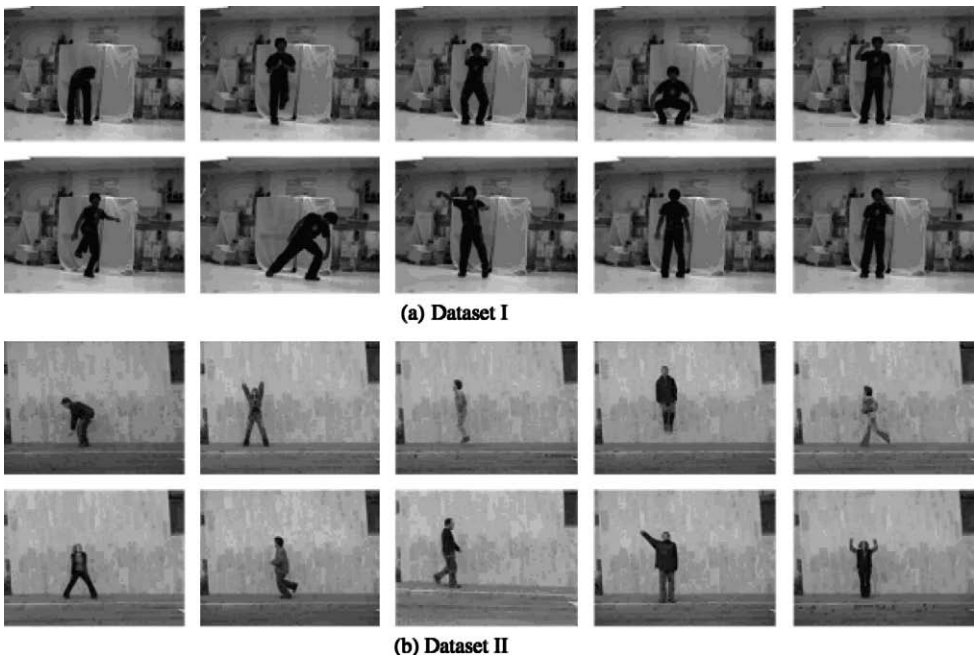

Fig. 5. Example images of action data.

Table 1
Recognition rates comparison for Dataset I consisting of 100 action sequences (%).

Actions	NN	SVM	GP	GP_ARD	GP_FO (with 50% of the features)
Pick	80.0	90.0	98.0	98.0	99.0
Jog	100.0	100.0	100.0	100.0	100.0
Push	90.0	100.0	100.0	100.0	100.0
Squash	90.0	100.0	100.0	99.0	100.0
Wave	100.0	100.0	100.0	100.0	100.0
Kick	30.0	70.0	98.0	96.0	95.0
Bend-side	100.0	100.0	100.0	100.0	100.0
Throw	70.0	100.0	99.0	99.0	99.0
Turn	100.0	100.0	100.0	100.0	100.0
Phone	80.0	100.0	95.0	100.0	96.0
Average	84.0	96.0	99.0	99.2	98.9

1996) for GP classification. GP_FO (GP with feature optimized) is the GP classification using feature reduction as described in Section 4.2 which retain only half (50%) of the original features. GP_FO implements the feature reduction on each LOO training set and applies the estimated parameters on the GP classification for the corresponding testing set. NN and SVM are also implemented as baseline experiments.

5.2. Human action classification results and analysis

We run Lawrence's program (Lawrence et al., 2004)[3] for GP classification. The results using standard GP, GP_ARD and GP_FO as well as comparison with the results using NN and SVM in (Wang et al., 2008) on the two datasets are summarized in Tables 1 and 2. Binary GP classification is used here for any of the actions by labeling the target action being +1 and all the rest of the actions being −1 in the training data and for classifying the test data.

It can be seen from Tables 1 and 2 that the three GPs have similar results and all perform better than either NN or SVM on both datasets. In addition, performance of GP_FO is almost the same as GP (GP_FO slightly worse than GP on Dataset I and a little better than GP on Dataset II) although GP_FO uses only half of the total features as used by GP. GP_ARD has slightly better results than GP for Dataset I and worse than GP on Dataset II. Fig. 6 shows that the computational complexity of GP_ARD is much higher than the rest of the GPs. GP_FO has the least computational time as a result of fewer features used.

It should be noted that although the overall classification rate of all GP-based approaches (GP, GP_ARD and GP_FO) is higher than

Table 2
Recognition rates comparison for Dataset II consisting of 198 action sequences (%).

Actions	NN	SVM	GP	GP_ARD	GP_FO (with 50% of the features)
Bend	88.9	100.0	100.0	98.5	97.0
Jump	87.0	95.7	94.1	96.5	95.5
Pjump	79.2	87.5	86.2	88.9	90.5
Run	59.3	92.6	90.0	92.0	94.0
Side	64.3	86.7	98.4	96.0	96.0
Kick	95.5	86.4	99.0	97.0	97.5
Skip	60.0	84.0	91.5	88.0	89.0
Walk	100.0	100.0	98.2	98.5	98.0
Wave1	89.5	100.0	99.5	98.5	98.0
Wave2	68.4	89.4	93.4	94.5	96.5
Average	78.3	91.4	95.0	94.8	95.2

[3] http://www.cs.man.ac.uk/neill/ivm/downloadFiles/.

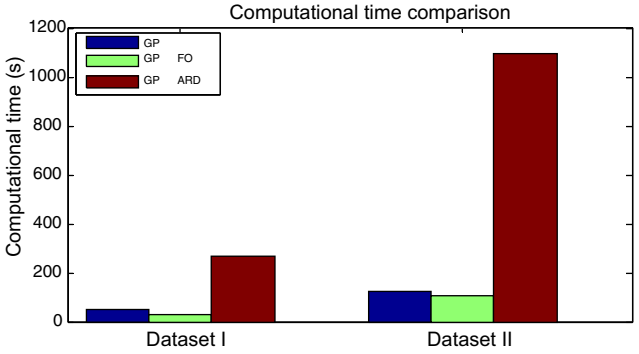

Fig. 6. Computational time comparison for Dataset I and Dataset II using GP, GP_FO and GP_ARD (s).

that of SVM, part of GP-based results are slightly worse than SVM. We believe this is because the GPs implement binary GP classification (it is much cheaper compared with the complicated multi-class GP which is beyond the investigation of this paper). Binary classifier is suboptimal for a multi-class recognition problem. The GP model is optimized on each LOO training set individually: this is another factor that causes suboptimal results.

By discarding the inconsistent features using spectral feature reduction, the classification results are almost the same as the standard GP and the calculation times are shortened. GP_ARD can also achieve similar results as the standard GP. However, the computational time is much longer on more than 200 features high dimensional datasets (more than 200 features) like Dataset I and Dataset II.

5.3. Comparison experiments

The feature reduction approach is also applied to the data used by Wang et al. (2008) and the results are compared with that of his Gaussian process dynamical models (GPDM).

As described in (Wang et al., 2008), the GPDM is used to learn human motion pose and motion from high dimensional motion capture data. A GPDM is a latent variable model comprises a mapping from a latent space x to the observation space y and a dynamical model in the latent space, with both learned from GP regression. The former is modeled by the RBF kernel:

$$k_Y(x,x') = \exp\left(-\frac{\beta_1}{2}\|x-x'\|^2\right) + \beta_2^{-1}\delta_{x,x'}, \tag{4.14}$$

where β_1 is the inverse width of the RBF kernel and β_2^{-1} is the additive noise variance. The signal-to-noise ratio (SNR) is

$$SNR(\bar{\beta}) = \sqrt{\beta_2}. \tag{4.15}$$

The latent space dynamic mapping uses a "linear+RBF" kernel, i.e.

$$k_X(x,x') = \alpha_1 \exp\left(-\frac{\alpha_2}{2}\|x-x'\|^2\right) + \alpha_3 x^T x' + \alpha_4^{-1}\delta_{x,x'}, \tag{4.16}$$

where α_1 is the RBF process variance, α_2 is the inverse width of RBF kernel, α_3 is the linear process variance and α_4^{-1} is the noise variance. The corresponding SNR is given by

$$SNR(\bar{\alpha}) = \sqrt{(\alpha_1+\alpha_3)\alpha_4} \tag{4.17}$$

$\bar{\alpha}$ and $\bar{\beta}$ are the model parameters and are estimated through learning the GPDM from the high dimensional motion capture training data. The learned model represents a distribution over new motions (poses) as well as their latent coordinates which should be similar to the training data and the corresponding latent coordinates. Hence, $SNR(\bar{\alpha})$ and $SNR(\bar{\beta})$ are reflections of the GPDM learning accuracy.

Table 3
Comparison of learned model using GPDM on the original data (GPDM) and feature-reduced data (GPDM_FO).

	GPDM	GPDM_FO
Number of features	50	25
α_1	0.02821078	0.03035969
α_3	0.48071395	0.41541685
α_4^{-1}	0.00011692	0.00011033
$SNR(\bar{\alpha}) = \sqrt{(\alpha_1+\alpha_3)\alpha_4}$	65.98	63.56
β_2^{-1}	0.00005933	0.00008108
$SNR(\bar{\beta}) = \sqrt{\beta_2}$	129.83	111.06
Computational time (s)	133	123

We run Wang's code[4] which covers learning a single walker model from data comprising two walk cycles from a single subject.[5] The single walker data is from Carnegie Mellon University motion capture (CMU mocap) database, where each pose is 50-dimensional, i.e. defined by 50 feature values.

To apply feature reduction, the latent coordinates are initialized using a subspace projection onto one principal direction given by PCA applied to the given data (which is already mean subtracted). Then, take the 50-dimension pose feature values as input and the projected one-dimension latent coordinate as output, NDFT and feature reduction are applied as described in Sections 4.1 and 4.2 respectively. The feature-reduced data is then learned with GPDM.

Table 3 shows the learned parameters using GPDM on the original data (GPDM) as well as the feature-reduced data (GPDM_FO). It can be seen that with the number of features reduced from 50 to 25, the corresponding SNR values of the two mappings of GPDM, i.e. $SNR(\bar{\alpha})$ and $SNR(\bar{\beta})$ only have a minor drop while the computational time shortened. SNR value depends mainly on the variance of the additive process noise. A higher $SNR(\bar{\alpha})$ value implies a smoother learned latent trajectories and a higher $SNR(\bar{\beta})$ value relates to a more realistic pose reconstruction. Therefore, by cutting half number of the features, only a minor change incurred to $SNR(\bar{\alpha})$ and $SNR(\bar{\beta})$ values which indicates that the accuracy of GPDM learning is still well maintained. In addition, the calculation is more efficient.

6. Conclusions

This paper establishes that GP classification can be an effective way to perform human action classification. We have shown that on average, GP's outperform NN and even SVM-based approaches. Moreover, we show how to analyse the data with a view to discarding features which makes the performance of GP even faster and still maintain the performance accuracy (on average). This last result is in contrast to the "traditional" way to perform parameter training and feature selection in the GP literature – ARD. In contrast to ARD, we generally perform better and at much cheaper cost. Comparison tests have also been done on the data used in (Wang et al., 2008) which shows that our feature reduction approach also works well on GP regression (in addition to GP classification) and applicable to varied data.

It should be noted that the proposed feature reduction method is essentially to choose the proper feature space, which is one of the fundamental problems of supervised learning, i.e. choosing the proper feature space and recovering the class membership function. Therefore, the feature reduction approach in this paper not only works on human action recognition, but also can be extended to other GP learning applications which are to learn from the data samples in a feature space and recover the class member-

[4] http://www.dgp.toronto.edu/~jmwang/gpdm/.
[5] CMU database file 07_01.amc, frames 1 to 260, down sampled by a factor of 2.

ship function, where the feature space need to be made more isotropic by cutting off some features so as to be better modeled by the isotropic GP kernel.

References

Bagchi, S., Mitra, S.K., 1999. The Nonuniform Discrete Fourier Transform and Its Applications in Signal Processing. Kluwer Academic Publishers.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R., 2005. Actions as space–time shapes. In: Internat. Conf. on Computer Vision (ICCV 2005).

Brand, M., Loiver, N., Pentland, A., 1996. Coupled hidden Markov models for complex action recognition. In: Internat. Conf. on Computer Vision and Pattern Recognition (CVPR 1996).

Bretthorst, G.L., 2000. Nonuniform sampling: Bandwidth and aliasing. In: Maximum Entropy and Bayesian Methods in Science and Engineering, pp. 1–28.

He, X., Cai, D., Niyogi, P., 2005. Tensor subspace analysis. In: Annual Conf. on Neural Information Processing Systems (NIPS 2005).

Lawrence, N.D., Platt, J.C., Jordan, M.I., 2004. Extensions of the informative vector machine. In: Deterministic and Statistical Methods in Machine Learning.

Neal, R.M., 1996. Bayesian Learning for Neural Networks. Springer-Verlag, New York.

Nguyen, N., Phung, D., Venkatesh, S., Bui, H., 2005. Learning and detecting activities from movement trajectories using the hierarchical hidden Markov models. In: Internat. Conf. on Computer Vision and Pattern Recognition (CVPR 2005).

Raskin, L., Rudzsky, M., Rivlin, E., 2007. Tracking and classifying of human motions with Gaussian process annealed particle filter. In: Asian Conf. on Computer Vision (ACCV 2007), vol. I, pp. 442–451.

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. The MIT Press.

Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D., 2005. Conditional models for contextual human motion recognition. In: Internat. Conf. on Computer Vision (ICCV 2005), vol. 2, pp. 1808–1815.

Snelson, E., 2006. Tutorial: Gaussian Process Models for Machine Learning. Gatsby Computational Neuroscience Unit, UCL.

Veeraraghavan, A., Chellappa, R., Roy-chowdhury, A., 2006. The function space of an activity. In: Internat. Conf. on Computer Vision and Pattern Recognition (CVPR 2006).

Wang, X., Smith, K., Hyndman, R., 2006. Characteristic-based clustering for time series data. Data Mining Knowl. Discov. 13 (3), 335–364.

Wang, J., Fleet, D., Hertzmann, A., 2008. Gaussian process dynamical models for human motion. Pattern Anal. Machine Intell. 30 (2), 283–298.

Wang, L., Wang, X., Leckie, C., Ramamohanarao, K., 2008. Characteristic-based descriptors for motion sequence recognition. In: The Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2008).

Zhou, H., Suter, D., 2008. Improving Gaussian processes classification by spectral data reorganizing. In: Internat. Conf. on Pattern Recognition (ICPR 2008).

Zhou, H., Wang, L., Suter, D., 2008. Human motion recognition using Gaussian processes classification. In: Internat. Conf. on Pattern Recognition (ICPR 2008).